

AMENDMENTS TO THE SPECIFICATION

Please replace the title with the following rewritten title:

A1
SYSTEM[[,]] AND METHOD, ~~AND ARTICLE OF MANUFACTURE~~ FOR LISTING
ACTIVITIES IN A GRAPHICAL USER INTERFACE IN A COLLABORATIVE WORK TOOL
FRAMEWORK

Please replace the indicated paragraphs with the following rewritten paragraphs:

A2
Page 14, lines 7-25

A preferred embodiment of a system in accordance with the present invention is preferably practiced in the context of a personal computer such as an IBM compatible personal computer, Apple Macintosh computer or UNIX based workstation. A representative hardware environment is depicted in Figure 2, which illustrates a typical hardware configuration of a workstation in accordance with a preferred embodiment having a central processing unit **210**, such as a microprocessor, and a number of other units interconnected via a system bus **212**. The workstation shown in Figure 2 includes a Random Access Memory (RAM) **214**, Read Only Memory (ROM) **216**, an I/O adapter **218** for connecting peripheral devices such as disk storage units **220** to the bus **212**, a user interface adapter **222** for connecting a keyboard **224**, a mouse **226**, a speaker **228**, a microphone **232**, and/or other user interface devices such as a touch screen (not shown) to the bus **212**, communication adapter **234** for connecting the workstation to a communication network **235** (e.g., a data processing network) and a display adapter **236** for connecting the bus **212** to a display device **238**. The workstation typically has resident thereon an operating system such as the Microsoft Windows NT or Windows/95 Operating System (OS), the IBM OS/2 operating system, the MAC OS, or UNIX operating system. Those skilled in the art will appreciate that the present invention may also be implemented on platforms and operating systems other than those mentioned.

A3
Page 28, lines 15-29

Figure 7 is an illustration showing a graphical user interface **700** for conducting sessions in a collaborative work tool architecture, in accordance with an embodiment of the present invention. The graphical user interface **700** includes session selection tabs **702**, a list of sessions **704**, wherein

A3
~~each the list of sessions 704~~ includes ~~a list of sessions 706~~. As discussed previously, each session 706 includes an activity start date 708, ~~[[a]]~~ an activity start time 710, ~~a session a session~~ duration 712, an activity title 714, and an activity status 716. In addition, a facilitator 718 is listed for each activity.

In use, a user may elect to participate in a session ~~706 704~~ by selecting a reply button 720 using a computer selection device, such as a mouse. In this manner, users may participate in multiple sessions simultaneously. Moreover, individual users may selectively indicate which sessions they will participate in and facilitator users may select which users to invite to particular sessions. As discussed in greater detail previously, the present invention determines a status 716 for each activity as the sessions proceed.

A4
Page 34, lines 15-21

Figure 13 is an illustration showing a graphical user interface 1300 for reporting in a collaborative work tool architecture, in accordance with an embodiment of the present invention. The graphical user interface 1300 includes a report window 1302 which displays a session report having a session title 1304, a session description 1306, a session creator 1308, and a session start and end time 1310. In addition, the report window 1302 further includes a list of the activities 1312 that occurred during the session.

A5
Page 42, line 8-Page 43, line 3

The best way to depict the process flow and the coordination of functions between each other is with the five flowcharts illustrated in Figures 16 to 20. Figure 16 depicts the overall process flow in accordance with an embodiment of the present invention. Processing commences at the top of the chart at function block 1600 which launches when the program starts. Once the application is started, the command line is parsed to remove the appropriate meeting text to initiate the target of the background find operation in accordance with an embodiment of the present invention as shown in function block 1610. A global stop list is generated after the target is determined as shown in function block 1620. Then, all the patterns that are utilized for matching operations are generated as illustrated in function block 1630. Then, by tracing through the chart, function block 1600 invokes GoBF 1640 which is responsible for logical processing associated with wrapping the correct search query information for the particular target search engine (function blocks 1650-1697). For example, function block 1640 flows to function block

A5 1650 and it then calls GoPatternMatch as shown in function block 1660. To see the process flow of GoPatternMatch, we swap to the diagram titled "Process Flow for BF's Pattern Matching Unit."

Page 43, lines 16-26

Search ALTA VISTA (~~Function block 2070 of Figure 20~~):

A6 As described in more detail herein, the ~~The~~ Alta Vista search engine utilizes the ~~identifies identities~~ and returns general information about topics related to the current meeting as shown in ~~function block 270 of Figure 2~~ Figure 20. The system in accordance with an embodiment of the present invention takes all the keywords from the title portion of the original meeting text and constructs an advanced query to send to Alta Vista. The keywords are logically combined together in the query. The results are also ranked based on the same set of keywords. One of ordinary skill in the art will readily comprehend that a date restriction or publisher criteria could be facilitated on the articles we want to retrieve. A set of top ranking stories are returned to the calendaring system in accordance with an embodiment of the present invention.

Page 44, lines 7-22

A7 Figure 17 is a user profile data model in accordance with an embodiment of the present invention. Processing commences at function block 1700 which is responsible for invoking the program from the main module. Then, at function block 1710, a wrapper function is invoked to prepare for the keyword extraction processing in function block 1720. After the keywords are extracted, then processing flows to function block 1730 to determine if the delimiters are properly positioned. Then, at function block 1740, the number of words in a particular string is calculated ~~and~~, at function block 1770 the delimiters for the particular field are checked, and a particular field from the meeting text is retrieved at function block 1750. Then, at function block 1780, the delimiters of the string are again checked to assure they are placed appropriately. Finally, at function block 1760, the extraction of each word from the title and body of the message is performed a word at a time utilizing the logic in function block 1762 which finds the next closest word delimiter in the input phrase, function block 1764 which strips unnecessary materials from a word and function block 1766 which determines if a word is on the stop list and returns an error if the word is on the stop list.

Page 51, lines 16-25

A8 Figure 19 shows a flowchart of the detailed processing for preparing a query and obtaining information from the Internet in accordance with an embodiment of the present invention.

Serial No 09/550,987

In reply to Office Action mailed May 23, 2003

Page 5 of 13

AS
Processing commences at function block **1900** and immediately flows to function block **1910** to process the wrapper functionality to prepare for an Internet search utilizing a web search engine. If the search is to utilize the Alta Vista search engine, then at function block **1930**, the system takes information from the meeting record and forms a query in function blocks ~~1940 to 1960~~ 1940 (including blocks 1942-1949), 1950, and 1960 for submittal to the search engine. If the search is to utilize the NewsPage search engine, then at function block **1920**, the system takes information from the meeting record and forms a query in function blocks **1921 to 1928**.
